



①9 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENT- UND
MARKENAMT

⑫ **Offenlegungsschrift**
⑩ **DE 199 26 580 A 1**

⑤① Int. Cl.⁷:
G 06 F 9/38
G 06 F 12/08

⑲ Aktenzeichen: 199 26 580.1
⑳ Anmeldetag: 11. 6. 1999
㉔ Offenlegungstag: 20. 1. 2000

DE 199 26 580 A 1

③① Unionspriorität:
09/115291 14. 07. 1998 US

⑦① Anmelder:
International Business Machines Corp., Armonk,
N.Y., US

⑦④ Vertreter:
Teufel, F., Dipl.-Phys., Pat.-Anw., 70569 Stuttgart

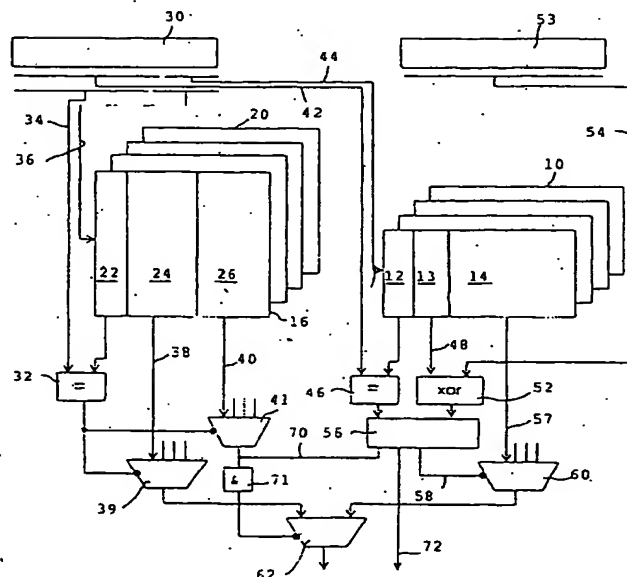
⑦② Erfinder:
Hilgendorf, Rolf, 71034 Böblingen, DE; Nair, Ravi,
Briarcliff Manor, N.Y., US

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Prüfungsantrag gem. § 44 PatG ist gestellt

⑤④ Verfahren und Vorrichtung zur Vorhersage von Sprungzieladressen

⑤⑦ Die Erfindung betrifft ein Verfahren und eine Vorrichtung zur Verkürzung von Rechenzeit während des Programmablaufs in elektronischen Rechnern als Maßnahme zur zeitlichen Anpassung der relativ langsamen Hauptspeicherzugriffe im Verhältnis zur schnelleren Verarbeitung der durch diese Zugriffe verfügbaren Daten durch den Prozessor und insbesondere die Vorhersage von Sprungzieladressen bei der Verarbeitung von Verzweigungsbefehlen. Es wird vorgeschlagen, in einem Cachespeicher (10) ein zweites Tag-Feld (13) mit zusätzlichen History-Informationen einzurichten, wodurch mehrere Sprungzieladressen pro Verzweigungsbefehlsadresse speicherbar sind. So kann ein Target Cache in einen BTB integriert werden.



DE 199 26 580 A 1

Die Erfindung betrifft Verfahren und Vorrichtung zur Vorhersage von Sprungzieladressen bei der Verarbeitung von Verzweigungsbefehlen.

In modernen Rechnersystemen müssen Maßnahmen getroffen werden, um die relativ langsamen Hauptspeicherzugriffe an die im Verhältnis dazu schnelle Verarbeitung der durch diese Zugriffe verfügbaren Daten in dem bzw. den Prozessor(en) anzupassen.

In Rechnersystemen mit superskalärer Prozessorarchitektur und Pipelinetechniken werden gleich mehrere Anweisungen aus dem Programm für eine Verarbeitung durch den bzw. die Prozessoren vorgehalten und verarbeitet. Folgen die Anweisungen sequentiell und unabhängig voneinander im Programm jeweils aufeinander, ergeben sich dabei keine Probleme. Komplizierter ist die Lage, wenn Verzweigungsbefehle (for, if, switch, etc. . .) in einer Pipeline verarbeitet werden sollen. Dabei sollte neben dem Ergebnis einer Verzweigung auch deren Sprungzieladresse bekannt sein. Verzweigungen sind hierfür sinnvollerweise unterscheidbar nach bedingten und unbedingten, sowie nach direkten und indirekten Verzweigungen.

Die Zieladressen sind für direkte Verzweigungen immer dieselben, meist durch ein Offset der Adresse des Verzweigungsbefehls selbst gegeben. Hierfür ist der Verzweigungszielspeicher (BTB, branch target buffer) ein Speicher, der für jeden Verzweigungsbefehl dessen Sprungzieladresse gespeichert hat, ein wirksames Mittel, um diese zum Zeitpunkt des Abrufs vorherzusagen.

Die Existenz von indirekten Verzweigungsbefehlen, die abhängig von den aktuellen Parametern des Programms, also dynamisch zur Programmlaufzeit zu verschiedenen Sprungzieladressen verzweigen, erhöht den Aufwand dafür, die richtige Sprungzieladresse vorherzusagen, da die aktuelle Parameterbelegung zum Zeitpunkt des Ladens der Pipeline mit den Anweisungen noch gar nicht feststeht. Während der Laufzeit erzeugte Sprungzieltabellen (branch history tables) stellen hierfür ein mögliches Mittel dar.

Das bedeutet, daß die Sprungzieladresse einer indirekten Verzweigung jedesmal eine andere sein kann, wenn sie ausgeführt wird. Daher ist es schwierig, die richtige Sprungzieladresse vorherzusagen, selbst wenn die Richtung der Verzweigung, d. h., ob sie erfolgt, d. h. springt oder nicht, bekannt ist. Ein Verzweigungszielspeicher (BTB) wäre hierfür nicht das geeignete Mittel, da dieser meist nur die zuletzt angesprungene Sprungzieladresse gespeichert hält.

Bisher wurde zur besseren Vorhersage von Sprungzieladressen beispielsweise in Chang, P.Y., Hao, E., Patt, Y.N. in "Predicting Indirect Jumps Using a Target Cache", Proc. 24th Ann. International Symposium On Computer Architecture, 1997, S. 274-283 vorgeschlagen, einen Cache-Speicher für die Sprungzieladresse zu verwenden, bei dem mehrere Adressen für ein und denselben Verzweigungsbefehl speicherbar sind, wobei Anordnungen mit bzw. ohne ein Tag-Feld vorgestellt werden. Das Tag-Feld enthält Informationen, die den Zugriff auf die richtige Sprungzieladresse über eine geeignete Indexbildung aus Adreß- und History-Informationen sicherer machen sollen.

Dabei wird die Adresse für den Zugriff auf den Cache-Speicher unter Verwendung der Adresse des Verzweigungsbefehls, wie es aus der BTB-Technologie bekannt ist, sowie unter Verwendung von History-Informationen erzeugt, die Informationen hinsichtlich der für den aktuellen Verzweigungsbefehl charakteristischen, vorher ausgeführten Befehlsadressen speichern und dazu verwendet werden, um zwischen den einzelnen Sprungzieladressen selektieren zu können.

Hierbei werden entweder sogenannte global pattern history oder alternativ dazu path history Informationen verarbeitet.

Die ersten verkörpern in Bitfolgen Informationen darüber, welche der letzten m Verzweigungen gesprungen sind und welche nicht, wobei nur die Abfolge gesprungen/nicht gesprungen gespeichert ist, nicht jedoch die zugehörigen Befehlsadressen. Dafür wird nur 1 Bit pro Verzweigungsbefehl benötigt.

Die letzteren (path history) verkörpern in Bitfolgen Informationen über die letzten n Verzweigungsbefehlsadressen selbst. Hierbei werden nicht vollständige Adressen, sondern meist nur die signifikanten, niederwertigen Bits, beispielsweise zwei Bits gespeichert.

Beide Typen von History-Informationen sind auf den gegenwärtigen Verzweigungsbefehl bezogen, für den eine Sprungzielvorhersage getroffen werden soll. Sie sind jedoch unterschiedlich dicht – ein bzw. zwei Bit pro Verzweigungsbefehl.

In dem oben erwähnten Verfahren müssen jedoch zur Vermeidung eines unverhältnismäßig großen Cache-Speichers bei der Anordnung ohne Tag-Feld die Adressen mit den History-Informationen einer Hash-Funktion unterworfen werden. Dies bringt den Nachteil mit sich, daß für unterschiedliche Verzweigungsbefehlsadressen und unterschiedliche History-Informationen auf den gleichen Sprungzieladressen-Eintrag zugegriffen werden könnte. Das bedeutet, daß für unterschiedliche Verzweigungen dieselbe Sprungzieladresse vorhergesagt würde.

Bei der Anordnung mit Tag-Feld im Cache-Speicher werden in der oben genannten Veröffentlichung drei verschiedene Methoden vorgestellt, wie das Tag-Feld gebildet und auf den Cache-Speicher zugegriffen werden kann. Dabei gelingt es nicht, auszuschließen, daß bei unterschiedlichen Verzweigungsbefehlsadressen und unterschiedlichen History-Informationen auf den gleichen Sprungzieladressen-Eintrag zugegriffen werden kann. Darüberhinaus besteht die Möglichkeit, daß für eine bekannte Verzweigung gar keine Sprungzieladresse vorhergesagt werden kann, wenn die Verzweigung noch nicht zusammen mit den aktuellen History-Informationen angetroffen wurde.

Aufgabe der vorliegenden Erfindung ist es daher, ein Verfahren und eine Vorrichtung zu schaffen, die die Auswahlmechanismen unter Vermeidung der zuletzt genannten Nachteile der fehlenden Eindeutigkeit der Sprungzieladresse zu verbessern und zu verallgemeinern.

Die Aufgabe wird durch das in Anspruch 1 angegebene Verfahren sowie die in Anspruch 8 angegebene Vorrichtung gelöst.

Die der vorliegenden Erfindung zugrundeliegende Idee besteht darin, daß eine Sprungzieladresse eines Verzweigungsbefehls dadurch vorhergesagt wird, daß auf denjenigen Sprungzieladressen-Eintrag im Cache-Speicher zugegriffen wird, der sich ergibt nach einem vorgeschalteten Abgleich von höherwertigen Bits der Verzweigungsbefehlsadresse, einem weiteren Abgleich von niederwertigen, verbleibenden Bits der Verzweigungsbefehlsadresse in einem ersten Tag-Feld des Cache-Speichers, wodurch zunächst die Identität des gespeicherten Verzweigungsbefehls mit dem aktuellen Verzweigungsbefehl sichergestellt wird, und einem nachfolgenden Vergleich auf möglichst weitgehende Übereinstimmung zwischen den in dem zweiten Tag-Feld gespeicherten History-Informationen mit denen der aktuellen History.

Das erfindungsgemäße Verfahren mit den Merkmalen des Anspruchs 1 und die erfindungsgemäße Vorrichtung mit den Merkmalen des Anspruchs 8 weisen gegenüber den vorstehend skizzierten Ansätzen den Vorteil auf, daß ein Ca-

che-Speicher für Sprungzieladressen (target-Cache) in einen herkömmlichen Verzweigungszielspeicher (BTB) integriert werden kann, wobei nur geringfügig mehr Fehlvorhersagen in Kauf genommen werden müssen. Desweiteren kann auch dann eine Sprungzieladresse vorhergesagt werden, wenn für einen bereits bekannten Verzweigungsbefehl eine noch unbekannte History vorliegt.

In den Unteransprüchen finden sich vorteilhafte Weiterbildungen des in Anspruch 1 angegebenen Verfahrens sowie der in Anspruch 8 angegebenen Vorrichtung.

Gemäß einer besonders bevorzugten Weiterbildung werden in dem zweiten Tag-Feld als History-Informationen bestimmte Bit-Folgen gespeichert, die aus einer XOR-Verknüpfung von global history und path history gewonnen werden, wodurch in den meisten Fällen die Vorhersagequalität noch weiter gesteigert werden kann.

Die nachfolgenden Ausführungsbeispiele der Erfindung dienen der Erläuterung des erfinderischen Prinzips. Sie sind in den Zeichnungen dargestellt und in der nachfolgenden Beschreibung näher erläutert.

Es zeigen:

Fig. 1 den schematischen Aufbau eines Eintrags einer integrierten Ausführungsform des erfindungsgemäßen Cache-speichers;

Fig. 2 in einer schematischen Gesamtübersicht eine Ausführungsform des Ablaufs des erfindungsgemäßen Verfahrens in einer weiteren Ausbildung des erfindungsgemäßen Cache-Speichers im Zusammenwirken mit einem separat angeordneten, herkömmlichen Verzweigungszielspeicher (BTB);

Fig. 3 in schematischer Darstellung einen Detailausschnitt der Ausführungsform aus Fig. 2.

Mit Bezug auf Fig. 1 wird zunächst eine zweckmäßige Ausgestaltung eines Eintrags des erfindungsgemäßen Cache-Speichers 10 erläutert, wie er in integrierter Form die Datenstruktur eines herkömmlichen BTB mitumfaßt.

Der Eintrag des Cache-Speichers 10 enthält ein erstes Tag-Feld 12, das eine bestimmte Anzahl von jeweils niederwertigen Adreßbits von Verzweigungsbefehlsadressen enthält. Im vorliegenden Ausführungsbeispiel, das auf eine Adreßbreite von 32 Bit ausgerichtet ist, enthält es beispielhalber die niederwertigen 24 Bit derjenigen Verzweigungsbefehlsadresse 30, für deren Verzweigungsziel die Vorhersage getroffen werden soll. Weiter ist erfindungsgemäß ein zweites Tag-Feld 13 vorhanden, das der zusätzlichen Differenzierung bei der Auswahl der Sprungzieladresse dient, wobei jedem zweiten Tag-Feld in einer n-fach Set-assoziativen Datenstruktur genau eine Sprungzieladresse in einem Feld 14 zugeordnet ist. Im vorliegenden Fall soll $n = 4$ gelten.

Ein weiteres Feld 16, das mehrere Bits umfaßt, ist für mehrere Flags vorgesehen. Hier können verschiedene Informationen bzgl. der Verzweigungen verschlüsselt sein, wobei die Flags beispielsweise Aussagen zulassen sollen, wie etwa:

ob eine bedingte oder eine unbedingte Verzweigung vorliegt,

ob eine identifizierte Returnadresse vorliegt oder nicht,

ob eine identifizierte Calladresse vorliegt oder nicht,

ob ein Verzweigungsbefehl mit mehreren, d. h. variierenden Sprungzieladressen vorliegt oder nicht, etc.

Durch Hinzunahme des zweiten Tag-Feldes 13 können erfindungsgemäß mehrere Sprungzieladressen für ein und dieselbe Verzweigungsbefehlsadresse 30 in dem Cache-Speicher 10 gespeichert werden, da deren eindeutige Zuordnung über das zweite Tag-Feld gegeben ist.

Mit Bezug auf Fig. 2 wird nun das erfindungsgemäße Verfahren im Zusammenhang mit dem erfindungsgemäßen

Cache-Speicher 10 in einer abgewandelten Form näher erläutert, wobei dieser in räumlich getrennter Form mit einem herkömmlichen Verzweigungsbefehlsspeicher 20 nach dem erfindungsgemäßen Verfahren zusammenwirkt.

Dem Verzweigungsbefehlsspeicher 20, der hier beispielhalber als 4-fach Set-assoziativer Speicher ausgebildet ist, ist für die Speicherung der verschiedenen Sprungzieladressen zum Zwecke der Verarbeitung eines Verzweigungsbefehls mit variierenden Sprungzieladressen der Cache-Speicher 10 als separater, kleiner und schneller Speicherbaustein nur für diesen Zweck zugeordnet. Ein solcher Aufbau kann sich insbesondere dann empfehlen, wenn eine geringfügig erniedrigte Trefferhäufigkeit nicht toleriert werden darf, oder wenn das Hinzufügen des zweiten Tag-Feldes für jeden Eintrag eine nicht mehr tolerierbare Flächenzunahme des Verzweigungsbefehlsspeichers 20 verursachen würde. In einem solchen Fall sollte die Assoziativität des Cache-Speichers 10 vorzugsweise zwischen 8 und 16 liegen, damit die meisten der möglichen Sprungzieladressen der Verzweigungen mit mehreren Sprungzieladressen in einen Set passen.

Wenn nun für einen aktuellen, in die Pipeline zu ladenden Verzweigungsbefehl eine Sprungzieladresse vorhergesagt werden soll, erfolgt in einem ersten Schritt in einem Komparator 32 ein Identitätsvergleich der höherwertigen Adreßbits der Verzweigungsbefehlsadresse 30 durch Zweig 34 und der niederwertigen, im ersten Tag-Feld 22 des Speichers 20 gespeicherten Adreßbits durch Zweig 36 jeweils mit den entsprechenden Adreßbitfolgen der aktuellen Verzweigungsbefehlsadresse 30, für die die Sprungzieladresse vorhergesagt werden soll. Das Ergebnis identisch/nicht identisch wird in eine Entscheidungslogik 39 weitergeleitet.

Bei Identität wird durch Abfrage des Flags 26 im Feld 16 des Verzweigungsbefehlsspeichers 20 über Zweig 40 und eine Entscheidungslogik 41 festgestellt, ob es sich um eine Verzweigung mit variierender Sprungzieladresse handelt oder nicht. Falls nicht, handelt es sich in den meisten Fällen um den einfachen Fall einer direkten Verzweigung, die den erfindungsgemäßen Cache-Speicher 10 gar nicht zur Vorhersage benötigt. Dann wird direkt die nach einer Initialisierung im Feld 24 des Verzweigungszielspeichers 20 gespeicherte Sprungzieladresse durch Zweig 38 und die Entscheidungslogik 39 zur Vorhersage bereitgestellt.

Falls eine variierende Sprungzieladresse vorliegt, wird zwecks Ermittlung der geeigneten Sprungzieladresse auf den Cache-Speicher 10 zugegriffen. Das Flag 26 kann nach zweimaligem Antreffen des Verzweigungsbefehls während der Initialisierungsphase geschrieben werden, da dann feststeht, ob sich die zugehörigen Sprungzieladressen unterscheiden haben oder nicht. Haben sie sich unterschieden, so wird das Flag 26 so gesetzt, daß auf den Cache-Speicher 10 zugegriffen werden soll. Der Zugriff kann dann beim nächsten Vorkommen des Verzweigungsbefehls über das geschriebene Flag 26 und den Zweig 40 erfolgen. Die weiteren, oben erwähnten Flags können ebenfalls über die Entscheidungslogik 41 verarbeitet werden.

Beim Zugriff auf den Cache-Speicher 10 wird analog zu dem oben geschilderten Abgleich der höher- bzw. niederwertigen Adreßteile des Verzweigungsbefehls sichergestellt, daß die im ersten Tag-Feld 12 gespeicherte Bitfolge den niederwertigen Bits der aktuellen Verzweigungsbefehlsadresse 30 entspricht.

Dieser Adressenvergleich findet im rechten Teil von Fig. 2 in den Zweigen 42, 44 und in dem Komparator 46 statt, mit der Maßgabe, daß die niederwertigen Bits aus dem ersten Tag-Feld 12 des Cache 10 stammen. Weiter findet über Zweig 48 der oben geschilderte Zugriff auf die zweiten Tag-Felder 13 des Cache 10 sowie im Verknüpfungspunkt 52 ein XOR-Vergleich der gespeicherten History-Informationen

auf Übereinstimmung mit den aktuellen History-Informationen 53 aus Zweig 54 statt. Dieser Vergleich ist weiter unten genauer beschrieben. In einer Entscheidungslogik 56 wird das zweite Tag-Feld 13 mit der bestmöglichen Übereinstimmung mit den aktuellen History-Mustern ermittelt. Die zugehörige Sprungzieladresse aus Feld 14 wird über die Multiplexsteuerung der Entscheidungslogik 60 als Auswahl aus dem 4-fach Set-assoziativ organisierten Speicher 10 über die Zweige 57 und 58 und die Entscheidungslogik 60 selektiert. Die Entscheidungslogik 62 gleicht, u. a. mit den Steuerbits aus dem Feld 16 ab und stellt die Sprungzieladresse für die weitere Verarbeitung zur Verfügung.

Mit ergänzendem Bezug auf Fig. 3, die einen Detailausschnitt aus Fig. 2 darstellt, der insoweit die Logikschaltungen 46, 52 und 56 umfaßt, wird der Zugriff auf den Cache-Speicher 10 und die Verarbeitung der History-Informationen darin näher beschrieben.

Nach dem Abgleich der Verzweigungsbefehlsadresse bei 46 wird in der Entscheidungslogik 52 der Inhalt der zweiten Tag-Felder 13 mit den entsprechenden Informationen der aktuellen History-Muster auf eine weitestgehende, beste Übereinstimmung verglichen. Dabei enthalten die zweiten Tag-Felder 13 vorzugsweise Bitfolgen als Informationsmuster bzgl. der Richtung der bisher vom Programm durchlaufenen Verzweigungsbefehle (global pattern history), oder bzgl. der Sprungzieladressen der zum aktuellen Verzweigungsbefehl führenden Verzweigungsbefehle (path history) des zugehörigen, durch die Informationen im ersten Tag-Feld beschriebenen Verzweigungsbefehls. Diesbezüglich wird für weitere Einzelheiten auf die in der Beschreibungseinleitung gemachten Erläuterungen Bezug genommen.

Es wird dann durch Vergleich diejenige Sprungzieladresse ausgewählt, deren zweites Tag-Feld am besten mit der aktuellen History-Bitfolge übereinstimmt. Diese Auswahl erfolgt auf besonders bevorzugten Weise durch einen Identitätsvergleich der Bitfolgen, von links mit den höherwertigen, der jüngeren Vergangenheit entsprechenden Bits beginnend und nach rechts zur älteren Vergangenheit hin fortschreitend. Die längste, übereinstimmende Bitfolge, die zweckmäßigerweise durch bitweise XOR-Verknüpfung mit den entsprechenden Bitfolgen der aktuellen History festgestellt werden kann, bestimmt über Zweig 58 durch die Multiplexsteuerung in der Entscheidungslogik 60 die vorherzusagende Sprungzieladresse. Diese Verfahrensschritte der bitweisen Vergleiche werden sämtlich in der Entscheidungslogik 56 durchgeführt, die beispielsweise als programmierbare Logikanordnung (PLA) implementiert sein kann.

In Fig. 3 sind die oben genannten, bitweisen XOR-Verknüpfungen durch Pfeile dargestellt. Nach rechts in der Figur fortgesetzt sind die Vergleiche mit den anderen, assoziativ zugeordneten Mitgliedern des Sets der zugehörigen History-Informationen der Verzweigungsbefehlsadresse dargestellt. Der Ausgangspfeil 70 signalisiert den Cache-Hit für den Fall, daß im Speicher 20 keine Sprungzieladresse für eine Vorhersage gefunden wurde. Der Pfeil 70 mündet in ein UND-Gatter 71 in Fig. 2, das mit den Steuerbits der Flags aus dem Feld 16 abgleicht. Der Pfeil 72 stellt die Verbindung zum Auswerteschritt, der Pfeil 58 stellt die Verbindung zur Multiplexsteuerung für die Auswahl der richtigen Sprungzieladresse dar.

Für den Fall, daß eine Sprungzieladresse falsch vorhergesagt wurde, was sich erst durch Abgleich des Ergebnisses des Auswerteschritts ergibt, sollte das die Sprungzieladressen speichernde Feld 14 des Cache-Speichers 10 aktualisiert werden.

Da für wird ein die Qualität – totale Übereinstimmung oder teilweise Übereinstimmung der verglichenen Bitfolgen – des XOR-Vergleichs wiedergebendes Ergebnis durch den

Zweig 72 an den Auswerteschritt weitergeleitet. Bei totaler Übereinstimmung wird die Sprungzieladresse im Feld 14, die zur falschen Vorhersage geführt hat, mit der neuen, vom Auswerteschritt gefundenen Adresse überschrieben. Bei teilweiser Übereinstimmung, was bedeutet, daß ein Eintrag mit den speziellen History-Informationen noch nicht existiert, wird im Cache-Speicher 10 das Feld 14, das am längsten nicht mehr verwendet wurde, mit der neuen Adresse überschrieben.

Das erfindungsgemäße Verfahren ermöglicht es im Zusammenhang mit dem erfindungsgemäßen Cache-Speicher 10, daß ein Cachespeicher für mehrere Zieladressen in einen herkömmlichen Verzweigungszielspeicher (BTB) integriert werden kann. Dabei müssen lediglich nur geringfügig mehr Fehlvorhersagen in Kauf genommen werden.

Gewöhnlich werden entweder History-Informationsmuster bezüglich der bisher vom Programm durchlaufenen Verzweigungsbefehle (Global Pattern) oder History-Informationsmuster bezüglich der Sprungzieladressen der zum aktuellen Verzweigungsbefehl führenden Verzweigungsbefehle (Path History Pattern) gespeichert.

Beide besitzen ihre spezifischen Vor- und Nachteile, die je nach Benchmarking zu unterschiedlich guten Ergebnissen führen.

Es hat sich jedoch herausgestellt, daß in den meisten Fällen eine XOR-Verknüpfung der beiden History-Informationsmuster trotz ihrer ungleichen Dichte und infolgedessen der bei gleicher Bitlänge unterschiedlichen Anzahlen m bzw. n der von ihnen beschriebenen Verzweigungsbefehle, vgl. Beschreibungseinleitung, zu den besten Vorhersage-Ergebnissen führt. Daher wird diese Kombination für eine Speicherung im zweiten Tag-Feld 13 des Cache-Speichers 10 als besonders bevorzugt vorgeschlagen.

Obwohl die vorliegende Erfindung anhand von bevorzugten Ausführungsbeispielen vorstehend beschrieben wurde, ist sie darauf nicht beschränkt, sondern auf vielfältige Weise modifizierbar.

Insbesondere können die Länge der erwähnten Tag-Felder je nach Zweckmäßigkeit variieren. Weiter sind Änderungen und Anpassungen des Verfahrens abschbar, wenn – durch die Eigenarten der jeweils dem Programm zugrundeliegenden Hochsprache oder durch Aufgabenstellung des Programms bedingt, oder aus anderem Grunde – die Häufigkeit von Verzweigungsbefehlen und die Anzahl der einem Verzweigungsbefehl zugeordneten Sprungzieladressen variiert.

Bezugszeichenliste

- 10 Cache-Speicher
- 12 erstes Tag-Feld
- 13 zweites Tag-Feld
- 14 Feld für Sprungzieladresse
- 16 Kennungsfeld
- 18 Vorhersage-Feld
- 20 Verzweigungszielspeicher
- 22 Tag-Feld f. niederw. Bits
- 24 Feld f. Sprungzieladresse
- 26 Flag
- 30 Verzweigungsbefehlsadresse
- 32 Komparator
- 34, 36, 38, 40, 42, 44, 48, 54, 58 Zweige
- 39 Entscheidungslogik
- 41 Entscheidungslogik
- 46 Komparator
- 52 Entscheidungslogik
- 53 History-Informationen
- 56 Entscheidungslogik, PLA

- 60 Entscheidungslogik
 62 Entscheidungslogik
 70 Zweig bei teilweiser Übereinstimmung
 71 UND-Gatter
 72 Zweig bei vollständiger Übereinstimmung

Patentansprüche

1. Verfahren zur Vorhersage von Sprungzieladressen bei der Verarbeitung von diesen Sprungzieladressen zugeordneten Verzweigungsbefehlen eines in einem ersten Speicher eines Rechners gespeicherten Programms, wobei Verzweigungsbefehlsadressen (30) zusammen mit diesen zugeordneten, aus dem Programmablauf gewonnenen History-Informationen zum Schreiben und Auswerten eines Tag-Felds in einem zweiten Speicher (10) geringerer Zugriffszeit als der des ersten Speichers verarbeitet werden, um einen selektiven Zugriff auf die vorherzusagende Sprungzieladresse zu ermöglichen, wobei bestimmte, den Verzweigungsbefehlsadressen (30) zuzuordnende Adreßinformationen in ein erstes Tag-Feld (12) des zweiten Speichers (10) geschrieben werden, **gekennzeichnet dadurch**, daß die History-Informationen in ein zweites Tag-Feld (13) des zweiten Speichers geschrieben werden und das Auswählen der vorherzusagenden Sprungzieladresse durch Abgleich des ersten Tag-Felds (12) mit den Adreßinformationen des Verzweigungsbefehls und durch Vergleich des Inhalts des zweiten Tag-Felds (13) mit den aktuellen History-Informationen (53) des zu verarbeitenden Verzweigungsbefehls erfolgt.
2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß der Vergleich bitweise erfolgt und die beste Übereinstimmung durch die Länge der übereinstimmenden Bitfolge festgelegt wird.
3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß auf die jeweils wahrscheinlichste Sprungzieladresse eines Verzweigungsbefehls ohne Auswertung der History-Informationen (53) an einem bestimmten Speicherplatz in einem integrierten Verzweigungszielspeicher (10, 20).
4. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß bei falscher Sprungzielvorhersage und vollständiger Übereinstimmung des Inhalts des zweiten Tag-Felds (13) mit den History-Informationen (53) des aktuell verarbeiteten Befehls die Sprungzieladresse des zugehörigen Eintrags im Verzweigungszielspeicher (10, 29) mit der richtigen Sprungzieladresse überschrieben wird und bei teilweiser Übereinstimmung vorzugsweise der am längsten nicht verwendete Eintrag überschrieben wird.
5. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß in dem zweiten Tag-Feld (13) jeweils global pattern History-Informationen und/oder path History-Informationen gespeichert werden.
6. Verfahren nach dem vorstehenden Anspruch, dadurch gekennzeichnet, daß in dem zweiten Tag-Feld (13) eine Bitfolge gespeichert wird, die sich aus einer XOR-Verknüpfung enthaltenden Verknüpfung der global pattern und der path History-Informationen ergibt.
7. Vorrichtung zur Durchführung des Verfahrens nach einem der vorstehenden Ansprüche, enthaltend den zweiten Speicher (10) als einen mehrere Einträge aufweisenden Cache-Speicher, der pro Eintrag das erste Tag-Feld (12), das zweite Tag-Feld (13) und ein Feld (14) für eine Sprungzieladresse enthält.
8. Vorrichtung nach dem vorstehenden Anspruch, da-

durch gekennzeichnet, daß der zweite Speicher (10) zusätzlich ein oder mehrere Kennungsfelder (16) zum Speichern von Steuerinformationen enthält.

9. Vorrichtung zur Durchführung des Verfahrens nach einem der vorstehenden Ansprüche 1 bis 6, dadurch gekennzeichnet, daß sie einen Verzweigungszielspeicher (20) enthält, der pro Verzweigungsbefehlsadresse (30) zur Speicherung von wenigstens einer, vorzugsweise genau, einer Sprungzieladresse eingerichtet und in integrierter, Form mit dem zweiten Speicher (10) nach Anspruch 8 vorgesehen ist, wobei wenigstens ein zusätzliches Kennungsfeld (26) zur Steuerung der Auswertung der in den Tag-Feldern (12, 13) vorhandenen Informationen vorgesehen ist.

10. Vorrichtung nach einem der vorstehenden Ansprüche 7 bis 9, dadurch gekennzeichnet, daß der zweite Speicher (10) als Cache-Speicher zur Speicherung von mehreren Sprungzieladressen mit einer Assoziativität von vorzugsweise 8 bis 16 ausgebildet und in separater Anordnung zum Zusammenwirken mit dem Verzweigungszielspeicher (20) eingerichtet ist, der seinerseits für jede Verzweigungsbefehlsadresse eine Sprungzieladresse sowie Steuerinformationen (26) enthält.

Hierzu 2 Seite(n) Zeichnungen

- Leerseite -

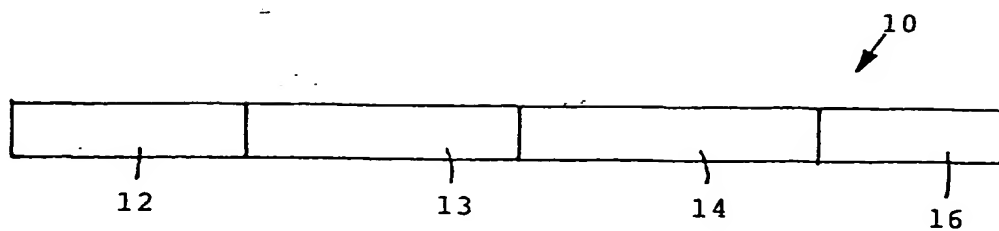


FIG. 1

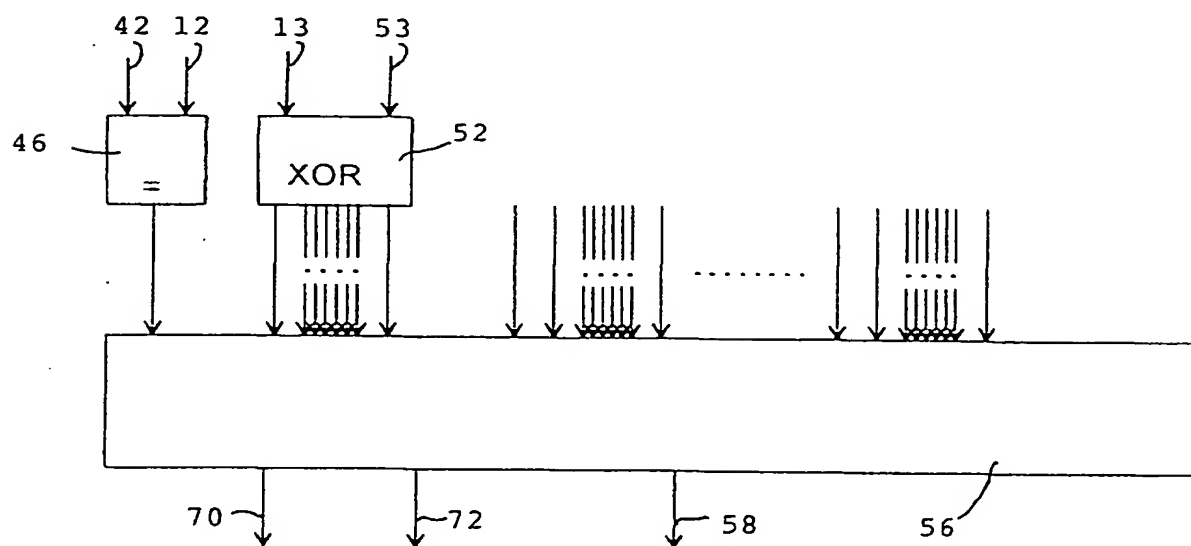


FIG. 3

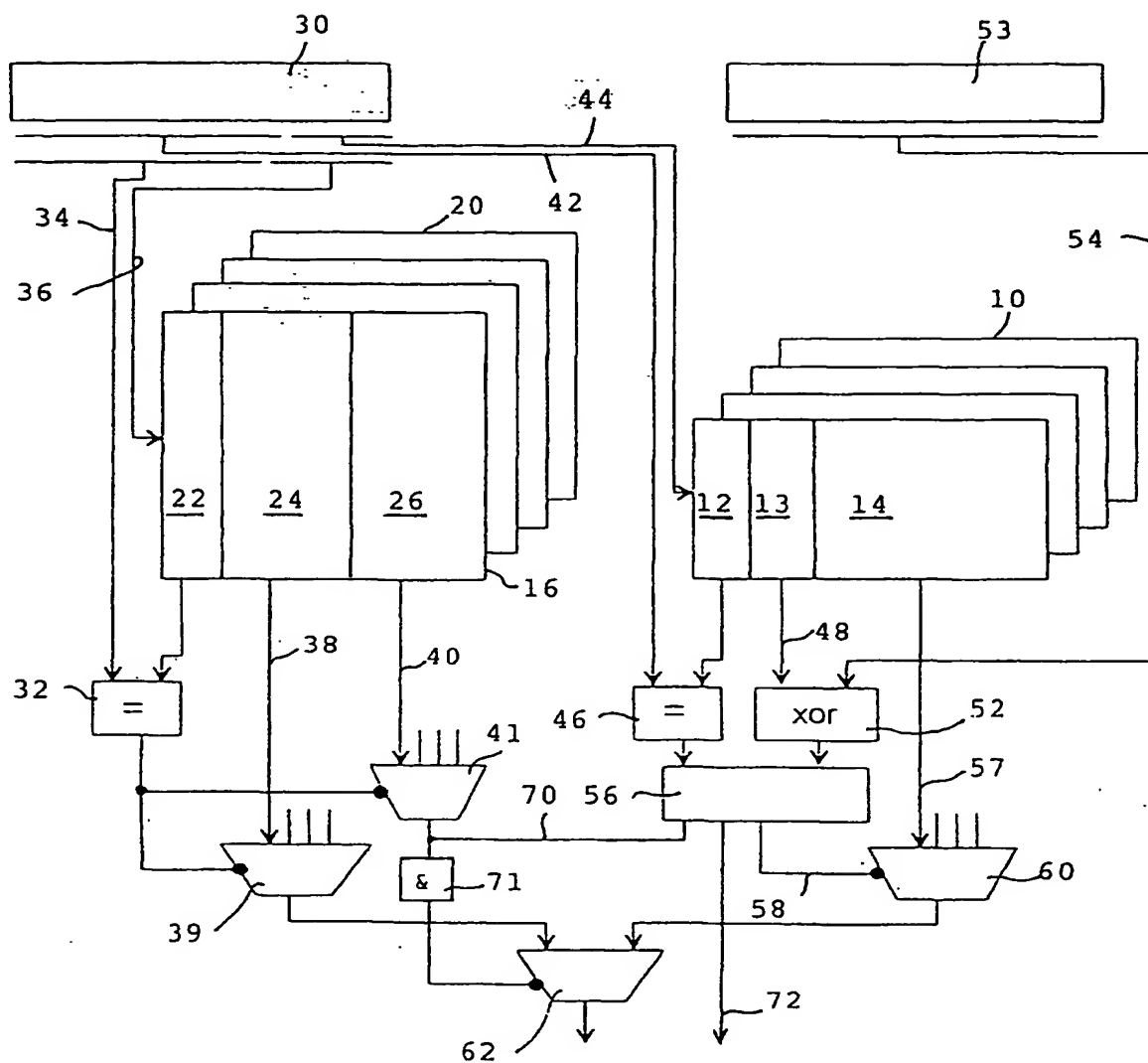


FIG. 2